ANALYZING AIR TRAFFIC MANAGEMENT SYSTEMS USING AGENT-BASED MODELING AND SIMULATION

A.P. Shah, A.R. Pritchett, K.M. Feigh and S.A. Kalaver, Schools of Aerospace Engineering and Industrial and Systems Engineering, Georgia Institute of Technology, (ashah, apritchett, kfeigh, kalaver)@isye.gatech.edu, Atlanta, GA

A. Jadhav, San Jose State University, amit_jadhav3@hotmail.com, San Jose, CA

D.M. Holl, R.C. Bea and A.Z. Gilgur, ATAC Corporation, (dmh, rcb, azg)@atac.com, Sunnyvale, CA

Abstract

This paper presents the viewpoint that an air traffic management system is emergent, i.e., exhibiting behaviors at the system-wide level that emerge from the combined actions of individuals within the system. Emergence carries with it the additional implication that these phenomena typically cannot be predicted by examining the individuals' behavior alone. As a result, this paper proposes agent-based simulation as a method of predicting the impact of revolutionary changes to an air transportation system. Agent based simulation can integrate cognitive models of human performance, physical models of technology behavior and description of their operating environment. Simulation of these individual models acting together can predict the result of completely new transformations in procedures and technologies. While agent-based simulations cannot include every aspect of system behavior, they can provide quick, cost-effective insights that can supplement other forms of analysis.

Introduction

In simulating air traffic management systems by agent-based simulations, we see them as emergent phenomena. 'Emergent' is formally defined here as a system property in which system behaviors at a higher level of abstraction are caused by behaviors at a lower level of abstraction which could not be predicted, or made sense of, at that lower level. More informally, our agent-based simulations are not based on any high-level models of an air traffic system; instead, we put agent models in a rich environment, simulate them in a realistic scenario, and see what system behavior results.

In this case our levels of abstractions are the agents (typically humans such as pilots and controllers) and the emergent system-wide behavior. Agent-based simulation provides interesting insights at both levels of abstraction. In addition to the system-wide behavior, in agent-based simulations the agents respond to their environment and each other. While we can model what the agents' responses would be to a variety of conditions, only simulation can predict what specific conditions they will need to respond to. As such, it is often just as interesting to use simulations to see what activities may be demanded of an individual agent when a revolutionary change is made to the air traffic system as it is to see what the system-wide behavior will be in response to changes in agents' capabilities or their environment.

This paper will discuss agent-based modeling and simulation methods suitable for simulating air traffic systems. These simulations can run autonomously for analysis purposes, but can also conceivably run in real-time for visualization and for human-in-the-loop interactions. In addition, such simulations can contribute to – and capitalize upon – other methods of research into human collaboration and coordination at both the 'micro' (individual) and 'macro' (system-wide) levels. These specific contributions and capitalizations will be noted throughout.

Background: Agent-based Simulation

The term *agent* has been used to mean anything between a mere subroutine or object and an adaptive, autonomous, intelligent entity [1-3]. This paper uses Hayes' definition of an agent as an entity with (1) autonomy, i.e., the capability to carry out some set of local operations and (2) interactivity, i.e., the need and ability to interact with other agents to accomplish its own tasks and goals [2].

Historically, agent based modeling concentrated on creating intelligent agents towards achievement of autonomy, an artificial intelligence perspective on emulating humans and designing autonomous technologies [3]. More recently, researchers have also applied "multi-agent" simulation of many interacting (but not necessarily fully autonomous) agents. Such multi-agent simulations have two concerns: modeling individual entities as autonomous and interactive agents, and simulating the system behavior that emerges from the agent's collective actions and interactions. These simulations are increasingly being applied in a wide range of areas including social sciences [4, 5], telecommunications, manufacturing [6, 7], business processes [8], and military simulations [9, 10].

One type of agent-based simulation has been used in purely social contexts to validate or illustrate social phenomena and theories or to predict the social behavior of interacting individual entities [11, 12]. Agent-based simulations of such social or natural systems often only include replications of one or a few homogeneous agent models. As such, use of such homogeneous agents does not allow for simulation of large-scale socio-technical systems (such as air traffic management) which may have agents fulfilling many roles.

Another type of multi-agent simulation has focused on "closed" systems, i.e., systems in which all aspects of the agents can be specified [13, 14]. These simulations employ complex heterogeneous agent models that can satisfy very different functions and roles, for example in teams of agents [9, 13]. Closed systems cover a number of applications, such as team activities, distributed sensor networks, and dedicated applications in which the goals of the agents can be aligned with the goals of the system [9, 13, 15].

In systems with multiple agents with differences in their beliefs, capabilities and desires, coordination and collaboration among agents becomes a necessity. Coordination refers to temporal management of action, events and tasks amongst agents. Coordination is usually achieved through communication of protocol or event information. Collaboration additionally requires agents to share their goals and intentions [16]. For multi-agent systems such communication is achieved through establishment of common semantics for those cognitive properties needing to be shared. Organizational structures, i.e., distribution of roles and tasks, might be imposed on the agents as an inherent mechanism for coordination [17].

Social primitives can be additionally employed to achieve collective behaviors from multi-agent systems [16, 18]. For example, agents may be designed to have mutual beliefs and joint intentions so that they inherently work towards same objectives and do not need to communicate as often [19].

Even though research in closed systems simulation has made significant accomplishments in simulating collective behaviors, closed system simulations are only good for scenarios that have well defined tasks and roles for a given set of agents. But air traffic management systems are of a more "open" nature. These systems satisfy a myriad of functions with agents manifesting different roles in different contexts. To achieve the desired system behavior from agents situated in rich contexts, one has to be able to tailor their work-environment [20-23].

Most recently, agent-based simulation has also examined the importance of explicitly modeling this work affecting aspect of the agents' environment and the agents' ability to sense and interact with it [24, 25]. This view draws on ideas from situation cognition, ecological psychology and cognitive engineering in which agents' behavior is seen as explicitly using and responding to their environment [20-23].

Air Traffic Systems as Agent-based

Building on the ecological perspective used in cognitive engineering, air traffic management systems exhibit the following characteristics, which make them suitable for agent-based simulation:

1. Involving a number of agents in a variety of roles and with a variety of and varying intents and capabilities;
2. Purpose or goal oriented;
3. Having a knowledge, culture, and established processes; and
4. Able to affect and be affected by their open environment.

The first characteristic identifies air traffic management systems as open agent-based systems. The agents themselves act to create the system behavior. They form a heterogeneous set whose variety (of roles) and variability (within roles) must be adequately captured. While some of their attributes can be specified (e.g., minimum criteria for training and selection), many attributes cannot: they will also have their idiosyncrasies, including individual goals and intent.

While the agents may only strive for their own goals, the second characteristic of air traffic management systems highlights that designers attempt to specify their “micro-level” behaviors to meet some system-wide “macro-level” purpose, such as specifying procedures for individual controllers and pilots that will create more efficient traffic flows. Agent-based simulation, unlike methods that focus on either micro- or macro-level behavior, simultaneously creates both. This allows not only examination of one or the other, but can also investigate the interplay between the two.

The third characteristic identifies aspects of air traffic simulations that need to be included in the models underlying the simulation. Agent-based simulation has the benefit of being *structure-preserving*, i.e., its model form and software implementation can (and should) mirror the structures of the “real” system. In the case of modeling air traffic, the agents and their environment can be represented in the simulation using the same semantics as the agents’ roles are defined to them in the system: identification of the agents and their environment, and description of their roles, duties, procedures and expected capabilities. This representation requires a level of effort commensurate with its implementation in the real system. For example, just as established air traffic procedures can be codified to be distributed to (or learned by) pilots and controllers, so can it be distributed to (or learned by) agents in a simulation. By using this representation, specific interventions can be examined, including changes to technology, organizational structure, procedures and information distribution.

The fourth characteristic mirrors the recent emphasis on explicitly representing the agents’ environment. This emphasis on an environment model allows for richer models of agents in which situated cognition, distributed cognition and expert adaptation to the environment can be explicitly captured. Likewise, in classic computational organization theory many environmental attributes are modeled as being distributed among (and coded into) the agents; therefore, these attributes do not extend beyond the lifetime of the agents [17]. In agent-based simulation the organizational, regulatory and physical aspects of the environment instead provide a structure in which agents can be embedded to generate a complete system behavior.

Thus, agent-based simulation captures well these characteristics of air traffic management systems. However, it can represent a deviation in practice from other methods of analyzing air traffic which incorporate patterns in system-wide behavior into their model. While the structure of the agents’ micro-level behavior is preserved in agent-based simulations, the macro-level behavior must be treated as emergent from the simulation, not pre-specified into the simulation. Likewise, it is typically not suited for focus on one facet of macro-level behavior alone, but instead creates the full system behavior with all its interacting and confounding aspects. In doing so, it provides a prediction of system behavior suitable for analyzing air traffic management concepts with a fidelity limited only by the validity of the agent models and of the structure within which they interact.

For example, an agent-based model of an air traffic management system may include:

- pilots and controllers as agents with cognitive human models;
- the physical environment as an assembly of airports, terrain, navigational aids and weather;
- aircraft as technological elements that dynamically interact with the physical environment and can be controlled by pilots; and
- traffic procedures as processes in the work environment.

These elements are listed here as an example. The specific choice of agents, and the fidelity with which they are modeled, can be tailored to the purpose and scope of the analysis.

Constructing Agent Based Simulations of Air Traffic

To construct an agent-based simulation of air traffic management systems, three components must be developed. First, models of the individual agents must be developed that are capable of emulating the relevant behaviors within the system. Second, a model of the environment must be developed which furnishes the agent models with the information they need about the physical and process aspects of their context. Third, mechanisms must be provided for the agents to act and interact, including mechanisms for timing the simulation and data passing within it.

These developments require both the conceptual models and their software instantiations. The conceptual models of the first two developments are tightly coupled with the domain being analyzed and capitalize on structure preserving abstractions. The third development is crucial to the architecture of the simulation engine and for the fidelity of the simulation as it governs the dynamics of the complete simulation. The software instantiations of the first two, while being verifiable and validate-able for the concepts they model, should also conform to the architecture of the third development.

Agent Models

As noted earlier, an agent is defined here as an entity having some autonomy in acting on its own as well as needing to interact with other agents in the system [2]. Implicit in this definition is the pro-activity of an agent – rather than only being a passive element of the system, an agent must act in ways that change the environment or the actions of other agents, and must interact with other agents [3].

These distinctions provide some guidance in identifying the agents in an air traffic management system. Passive elements incapable of changing the state of the environment or other agents are typically best described within the environment model, as later discussed. Pro-active elements that act completely autonomously do not contribute to an agent-based simulation involving interacting agents.

Within this guidance, selecting what entities should be modeled as agents within a simulation is not always clear-cut. Physical entities may be modeled as agents, or agents may be defined around functional attributes and tasks [24, 26]. Each agent may represent the behavior of one human in the system, or different agents may handle different tasks involving multiple humans. For example, each air traffic controller may be represented by an agent, or teams of controllers performing one function may be represented as an agent, or many controllers may be modeled with one agent for their 'monitoring' activities, one agent for their 'conflict resolution' activities, etc. The final selection of agents is an important design decision in developing an agent-based simulation that should be based on the purpose of the simulation and the required fidelity of each of the agent-level behaviors.

Modeling human performance is a common basis for many agent models in agent-based simulation of human integrated systems. Several different research communities have created a wide variety of such models: the artificial intelligence and intelligent systems community in computer science; the computational organization theory community; and the human performance model community in cognitive science and human factors.

Cognitive models suitable for agent-based simulation are increasing in detail and ability to capture relevant aspects of human performance, where human characteristics, based on empirical research, are embedded within a computer software structure to represent the human operator [27, 28]. These models are described as modeling *performance* rather the behavior because of their scope – the current state of the art is better at capturing purposeful actions of a human as generated by well-understood psychological phenomenon than it is at modeling in detail all aspects of human behavior not driven by purpose.

Simple forms of human performance models may use engineering models to replicate identifiable tasks. For example, Pritchett, Lee and Goldsman [29] modeled air traffic controllers as using simple "dead-reckoning" navigation filters to predict whether aircraft would lose safe separation and to determine speed commands to resolve such conflicts. While such models are often mechanistic and limited to specific tasks, they can capture well-established patterns of performance; for some applications of agent-based simulation this can be sufficient.

Other "selfish" model forms may view agents as pursuing their goals using optimization or decision making mechanisms to select their actions. (For an example in air transportation simulation, see [30]; for examples using Markov Decision Processes, see [31]). A specific form of agents capable of pursuing their own goals uses the cognitive primitives of beliefs, desires, goals, intentions and commitments [16]. In addition, selfish model forms can be used in simulations with "learning agents" with each agent remembering statistics over multiple runs, thus converging on an improved course of action to take in the next. For example, [30] detailed a simulation in which, given a fixed level of demand for air transportation and capacity at various airports, learning agents of airlines and passengers can be re-run hundreds or thousands of times to find the flight schedule and fare structure maximizing airlines' profits without exceeding airports' capacities for arrivals and departure.

Another model form uses normative models of performance, i.e., models based upon the prescribed processes that the agents should follow, as judged by the system designer or other external entity. The agent may be modeled as following its processes exactly, following them with some variation (e.g., with stochastic reaction times fitting observed human behavior), or in more complex manners such as selecting processes in forming intentions that will meet their goals. These processes may be codified into the agent model by rule-bases and expert system. Even when such agent models follow normative standards exactly and deterministically, agent-based simulation provides two interesting insights. First, when the environment is as expected, executing the simulation will reveal whether satisfactory emergent system behavior is created by agents who are all exactly acting according to procedure [32]. Second, the environment in which the agents operate can be perturbed to examine the impact of likely or potential failures due to normative behaviors, thus identifying conditions that require agents' discretion for violation of the norm to satisfy other objectives.

The most advanced models may include elements of each of these model forms. For example, Air MIDAS, developed by NASA Ames Research Center (ARC) and San Jose State University (SJSU) primarily for aviation-related applications, contains several functions within its model of human performance. Mechanistic models of essential psychological and physiological phenomenon such as vision, attention, working memory and motor skills capture well-understood aspects of human behavior. Domain knowledge serves as pre-established knowledge about the task, often represented as procedures and a rule-base of goals and processes for core tasks. An upgradeable world representation also acquires and maintains knowledge about the current state of the environment. Within this framework, a symbolic operator model maintains queues of tasks waiting to occur, and switches tasks between them according to knowledge and goals [27].

Environment Models

Including an environment model in an agent-based simulation requires a slightly different conception of "environment" than that commonly used in systems engineering. Rather than viewing the environment as everything outside the system boundary, in agent-based simulation the environment spans all the passive elements of the system that situate the functioning of the pro-active agents. By definition, the agents' environment can have a dramatic impact on their individual behavior and, as a result, on emergent system performance; its elements (including physical space, new technologies, and procedures and regulations) are often the means by which system-wide change is effected.

In agent based modeling the environment is seen with respect to the objectives and cognitive mapping of the agents. The objective representation of the environment is contained within the environmental model during the simulation, whereas the agents maintain subjective representations of the environment to govern their actions.

There are two elements to kinds of environmental models. The first models the physical environment. The most common structure models the physical surroundings of the agents through various spatial representations (for example topological maps for robot navigation [33]). In air traffic control simulations, the locations of navigation aids, airports etc. are described using commonly understood coordinate systems [32].

The second specifies particular aspects of the work domain. Many of these aspects structure not only objects in the environment but also tasks and their relationships. Each such structure captures one dimension of the environment; a combination of structures can create a multi-dimensional environment model for agents to reference. Several structures may be used:

- Work domain analysis introduced *structural means-end relations* [23]. These structural relationships are hierarchical, i.e., elements of the environment are arranged according to levels of abstraction connected by means-end relations: An element in one layer is an end that can be achieved by employing elements in the layer below, and a means to achieve the ends of elements in the layer above. This model can be utilized in selection of strategies in employing available resources.
- *Task dependence structure*s organize tasks into acyclic graphs called task groups [34]. The root node identifies the main task to be accomplished, and the sub-levels identify sub-tasks and methods that may be executed to accomplish the root task. This hierarchy is further enriched by task dependence relations between subtasks and methods such as enables, facilitates, disables, and delays.
- *Context-process structure*s defines the context of an agent as having two parts: world state and the agent's intention. [20, 21]. This structure complements cognitive models that can select between processes in situational context [25, 35].

Each of the structures discussed above describes and models one objective aspect of the environment. Many other structures are plausible, such as task-resource structures (relating resources to tasks) and task-skill structures (relating tasks to required skills) [23]. The same environmental element may be represented in a number of these structures and the concerns of any agent may cut across several of them, as suits the purpose of the analysis.

Simulation Architecture

Agent-based simulation requires two further developments beyond the conceptual specification of agent and environmental models: (1) codifying these conceptual models with sufficient exactitude that they can be implemented as computational software objects and (2) placing these software objects within a larger software architecture which creates and maintains their correct interactions.

In doing so, an important intellectual duality exits between the conceptual models' forms and the software architecture. Software engineering methods such as object-oriented programming can provide the much needed benefits of understandable, readily modified, easily re-used software. However, these benefits can only be fully realized if the software architecture is laid out well from its inception. This layout must mimic and support the conceptual models' form so that the translation between conceptual model and software implementation is fluid and conformal – to do so, the conceptual model must be specific and well defined.

Such conformal mapping between the conceptual model and the software implementation can have several benefits. First, many established systems have converged on efficient methods of operation; simulation software mimicking these operations has a better chance to be computationally efficient, without extraneous computations. Second, when decomposing the simulation spatially or temporally to focus on more specific aspects of the simulation, a natural decomposition strategy can be found that does not adversely impact the simulation's functioning. Third, the software's behavior can be easily verified relative to that expected conceptually without translating between or adjusting for different model inputs, outputs and behaviors.

Setting up such a simulation architecture requires three main considerations: the interface standards for the software objects, the method of advancing simulation time (and having agents interact at the correct time), and methods of interacting with the larger analysis process. The following three sections detail these considerations.

Software Interface Standards

Software interface standards specify the functions within software components and their methods for data passing, thus defining several aspects of an agent-based simulation architecture. The first aspect concerns the general functions that should be internal to the agent models. These generally should be only those that define the autonomous actions of the agents, and each agent's internal functioning during interactions with the environment and other agent. Second, the functions internal to the corresponding environmental model need to be defined. Third, standards need to specify which functions are executed by the simulation architecture (and, correspondingly, not executed by the agents and environment model); these generally reflect those functions arising from the simulation and analysis processes, including visualization, data recording, and time advance.

A fourth aspect concerns the data passing requirements of each component in the simulation, including which data each publishes to the rest of the simulation environment, and specifications of whether the components 'push' data to each other or 'pull' data from each other. When, in the real system, information transfer is clearly initiated by one entity and received by another, then a clear conceptual basis for the standard exists. Unfortunately, not all information transfer mechanisms are that clearly defined in socio-technical systems at the level of detail required for emulative agent models. For example, in reality an air traffic controller doesn't need to survey all aircraft in the world to identify those within his or her control sector; however, without special 'radar' models, an air traffic controller agent either needs to 'pull' from simulation all the aircraft positions to identify those under its control, or the aircraft need to 'push' their information to all controllers. Distinguishing these mechanisms is generally as much a conceptual effort in modeling the 'real' information transfer processes as a software development task.

Time Advance and Agent Interactions

The heart of every simulation is a timing mechanism that advances simulation time and selects the object to be executed next [36]. Time constitutes an important component in the behavior of the agents and their interactions; as such, timing mechanisms are one facet of modeling system dynamics [37]. In agent-based modeling and simulation, timing mechanisms need to address two main issues. First, timing mechanisms need to properly handle heterogeneous agent models, which may have considerably different update rates for their internal dynamics. Second, agents must be timely updated for correct interactions between agents.

Timing mechanisms can be typically defined as synchronous or asynchronous. While synchronous timing methods require all agents in the simulation to update at the same time, asynchronous timing methods allow each agent to update independently. For large-scale or repeated runs of the simulation, synchronous timing methods are usually computationally inefficient since the timing method requires all agents to update at every time step, whether each needs to or not. The timing method *asynchronous with resynchronization* [29] allows agents to update asynchronously following their own update times, but also estimates when interactions may occur in the future, and requires the relevant agents to jointly update at the times of interaction. To operate within a simulation architecture using this timing mechanism, each agent must be able to (1) remember the time of its last update, (2) report the next time it needs to update for its autonomous behaviors, and (3) update when commanded by the central timing mechanism.

Each agent can be endowed with the ability to monitor and predict its interactions with other agents. However, it can be very difficult and costly to endow agents with more accurate and complex predictive power in a simulation, especially once the simulation contains stochastic elements or when interactions arise due to emergent behaviors and are thus not easily predicted [38]. Instead of endowing each agent with the capability of precisely predicting the interactions with other agents, for many applications it is more natural and efficient to develop an object that can monitor and predict the interactions between agents. This object may be conceptualized as providing the monitoring behaviors and physical relationships within the system that trigger interactions between agents.

Case Study: Arrivals to LAX

The authors are currently employing agent-based simulations to demonstrate the application of agent-based simulation to risk analysis of arrival procedures. This study employs real world scenarios, which include differences in traffic flow control methods such as Time Based Metering (TBM) and Miles in Trail (MIT) from the Los Angeles International (LAX) airport. This study is specifically simulating the six eastern approach sectors, with arrivals, departures and over flights in a variety of weather conditions. Weather conditions of interest include wind direction (head or tail wind), wind speed (high or low) and visibility (which impacts allowable arrival rate). Several agents and aspects of the environment are modeled, as shown in Figure 1.

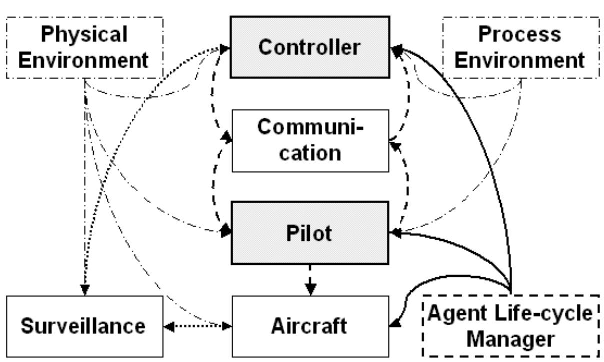


Figure 1. Agents and Environmental Aspects in Simulation of LAX Arrivals

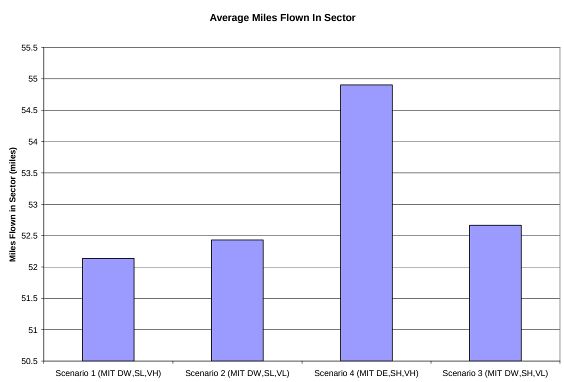
In this study the controllers are the primary objects of interest. Consequently they are the proactive elements that have been modeled to significant fidelity by employing the human behavior and performance models implemented in Air MIDAS [27, 28]. The controller agents are continuously updated by a dynamic environment model about the state of the airspace and the flight path information of each aircraft. The controller agent model continuously updates its internal representation based on the updates and cycles through its activity map determine its control actions. Each activity is modeled in ways that exerts a certain amount of activity load on the controller and occupies the controllers' cognitive resources. These activity maps can be selectively changed or altered in the agents' cognitive model as per the applicable procedure. Thus the simulation is capable of changing agent behavior to use an entirely different set of procedures (which would be a concern with human subjects).

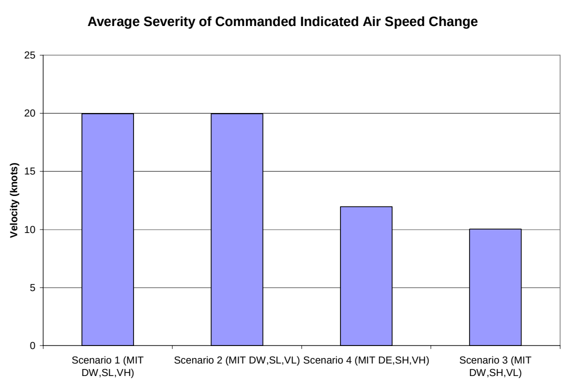
The controllers command the aircraft in a fashion similar to the real world, i.e. in a structure-preserving manner where each command is sent as per the air traffic control protocol. These commands are sent through a simulated frequency channel, which broadcasts the message to all aircraft tuned into that frequency. This mechanism emulates the sphere of influence of an air traffic controller, which is limited to aircraft on that frequency. This also emulates the sphere of perception of an aircraft pilot who can only hear the controller and aircraft talking on the frequency on which the pilot is tuned.

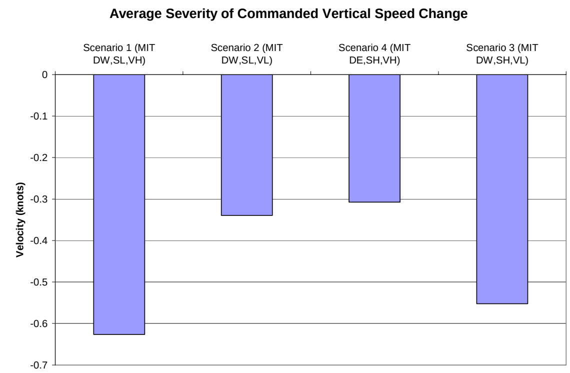
The pilot is modeled as an agent who can receive controller commands and can fly the aircraft through standard operating procedures. The pilot in this study is capable of deciding when it can or cannot comply with controller commands. But this intelligence has been limited to very specific decisions necessary for the ATC controlled approach control safety analysis. For example, the pilot can decide that it cannot intercept a given radial based on its perception of the physical context.

The aircraft are modeled to three degrees of freedom and account for the performance limitations of commercial aircraft. Their dynamic models interact in continuous time with their environment that provides them with the prevalent wind conditions.

Higher level of fidelity pilot and aircraft models were not used for this particular simulation for it was not considered necessary for the objectives of this analysis, but could be added without requiring changes to the architecture or controller agent model.







The pilot and other physical entities such as the aircraft, the navigational aids, the winds and the communication channels have been modeled in the Re-configurable Flight Simulator (RFS) [29]. RFS thus provides the physical environment for the controller and the pilot agents. It also provides other software architecture required to manage the life cycle of the aircraft and the pilots as they arrive and depart the airspace covered in the simulation. Aircraft are injected into the simulation at specific times following records of recent days at LAX; some of the arrivals will simulate particularly stressful patterns of traffic.

At the micro-level, controller workload and performance are being measured. Since human cognitive processes are modeled to their best known primitives, one does not depend on subjective perception, i.e., an estimated value of the associated workload, as may be estimated by subject matter experts. Moreover these models are able to take into account cognitive process load in relation to the particular situational context and not as a general random process that may or may not be valid for that specific context.

At the macro-level, the traffic flow patterns that results from the actions of the controllers and pilots are also being recorded. From these results, measures of arrival efficiency and safety can be made. (These simulation results are also being validated against real world scenarios from which input data was collected. This includes validation of flight paths and types of controller commands.)

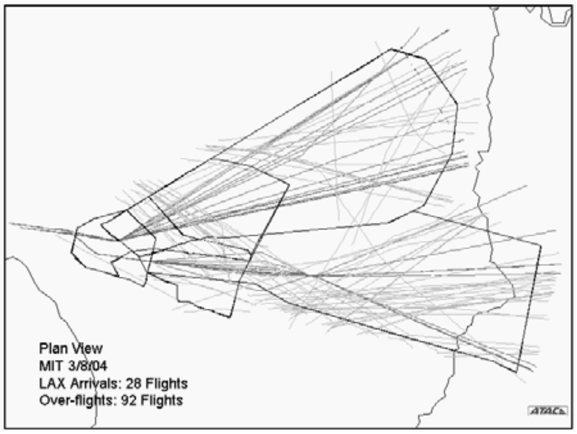


Figure 2. Simulated LAX Flights

Conclusions and Open Questions

Agent-based simulation is a comparatively new method for analyzing air traffic management systems. It builds on those aspects of air traffic management systems that can be directly observed or specified – the work practices of the agents themselves. From these, it predicts the behavior of the system as a whole, and the corresponding demands the environment will place on its agents, typically the pilots and controllers.

In doing so, this method can answer many specific questions about the systems it examines. With simple normative agent models, for example, agent-based simulation can observe whether the system will function as desired when all participants act exactly as procedures, regulations and organizational structures mandate – and highlight areas where individuals' flexibility and creativity are still required to operate the system. At this time, if we were to simulate a large air traffic system with all agents exactly following procedures, would our simulation results mimic actual practices? Until operating procedures are that specific (and consistently correct), we can not specify behavioral requirements for automation and the humans in the system are still contributing in a manner that may not be fully understood.

Likewise, such normative agents can also record anytime their required behavior conflicts with what they themselves would have done to meet their own goals, highlighting conditions where the real system would be susceptible to non-procedural (or unethical) behaviors. For example, a simulation of 'free flight' would highlight when pilots may be tempted to cut corners in a manner not desirous from the point of view of the overall traffic flow.

With more descriptive models of human behavior, the simulations could additionally serve to predict the outcome of variable and unintended behaviors. By simultaneously simulating the 'macro' level system-wide behavior and the 'micro' level agent level behavior, the work environment of agents is also captured, suitable for use in real-time human in the loop simulations or for computationally estimating the demands imposed on the agents.

This method also illustrates several more general questions whose answers may be subject of debate for some time to come. First, are air traffic management systems emergent? While answering this question comprehensively for all socio-technical systems would be difficult, even the few examples given here highlight insights from agent-based simulations which could be predicted a priori at this level of detail by no other method –for example, the change in air traffic control patterns brought on by the implementation of a new arrival spacing procedure and decision aid. Emergent behaviors appear to be the most relevant when a system may be reasonably judged by the product of both individual agents' autonomous actions and their interactions. This is especially relevant when the complexity of and uncertainty within the system obstructs each agent's awareness of the impact of their local behavior on overall system performance.

Second, what are the important aspects of air traffic management systems to include in simulations? Many models focus on specific aspects of system behavior. On the other hand, in forming agent-based simulations, accurate prediction of the high-level system behavior requires modeling of many aspects of its low-level structure. The agent models themselves have received the most attention in the research community, and can have a variety of different forms. In addition, given our own emphasis on ecological models of cognition, we argue that a rich environmental model is needed that defines the physical and social constraints on agent behavior, and that establishes the proper mechanisms for agent interactions.

Third, what are the design variables in air traffic management systems – and what are the emergent properties? The historic emphasis of agent-based simulation on analyzing and changing the agents themselves has highlighted the emergence of system behavior from agent behaviors. These insights alone can be useful; in addition it is important to note that, in dealing with the humans in air traffic management systems, we often can't change the human's cognitive properties directly; instead, we can change the physical and regulatory aspects of their environment, and we can change the procedures and technologies through which they interact. As such, these design variables can be made manifest in a structure-preserving manner in the environment model. An interesting side-effect of this viewpoint is that changes in the environment model can then create emergent behaviors in the higher-level system and adaptations in the lower-level agent behaviors as they respond and adapt to the demands of the environment.

Finally, given the time and expertise required to establish agent-based models and simulations, is this method practicable for analyzing air traffic management systems? This chapter has discussed how a close synergy between the conceptual modeling of agents, the conceptual modeling of air traffic management systems, and the software engineering concerns in establishing computer simulations can provide insight to each other, and thus establish a streamlined analysis and design process. The software engineering concerns center around making simulations that are re-usable and re-configurable. When the agent-based simulation architecture allows for agent and environment models that are structure preserving, the conceptual models should not require translation when codified into a computational representation. Ultimately, by representing the organizational environment in a form that can be specified in vivo to the people in the socio-technical system as well as the simulated agents, the simulation can serve as a design repository in which design variables are not only tested, but also stored and accessed during operations.

Acknowledgments

The authors are grateful for the support and direction of Dr. Irv Statler at NASA/Ames Research Center. The work reported here was partially funded by NASA under Grant NAG2-01291 to Georgia Institute of Technology. This on-going work is being conducted jointly with San Jose State University; the collaboration of Kevin Corker and Amit Jadhav is gratefully acknowledged.

References

[1] Franklin, S. and A. Graesser, 1996, *Is it an agent, or just a program? A taxonomy of autonomous agents*, in Third International Workshop on Agent Theories, Architectures and Languages.

[2] Hayes, C.C., 1999, *Agents in a nutshell - a very brief introduction*, IEEE Transactions on Knowledge and Data Engineering, **11**(1).

[3] Wooldridge, M., 2000, *Intelligent Agents*, in Multiagent Systems: a modern approach to distributed artificial intelligence, G. Weiss, Editor, The MIT Press.

[4] Gilbert, N. and K.G. Troitzsch, 1999, *Simulation for the social scientist*, Buckingham, Open University Press.

[5] Goldspink, C., 2000, *Modeling social systems as complex: Towards a social simulation meta-model*, Journal of Artificial Societies and Social Simulation, **3**(2).

[6] Barbuceanu, M., R. Teigen, and M.S. Fox, 1997, *Agent-based design and simulation of supply chain systems*, in 6th Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises (WET-ICE'97): Institute of Electrical and Electronics Engineers, MIT, Cambridge, MA.

[7] Shen, W. and D.H. Norrie, 1999, *Agent-based systems for intelligent manufacturing: A state-of-the-art survey*, Knowledge and Information Systems, an International Journal, **1**(2): p. 129-156.

[8] Huang, C.C., 2001, *Using intelligent agents to manage fuzzy business process*, IEEE Systems, Man and Cybernetics: Part A, Man and Systems, **31**(6): p. 508-523.

[9] Tambe, M., 1997, *Agent architectures for flexible, practical teamwork*, in American Association of Artificial Intelligence.

[10] Tambe, M., K. Schwamb, and P.S. Rosenbloom, 1995, *Building intelligent pilots for simulated rotary wing aircraft*, in Conference on computer generated forces and behavioral representation.

[11] Conte, R., R. Hegselmann, and P. Terna, 1997, *Simulating social phenomenon*, Berlin, Springer.

[12] Davidsson, P., 2002, *Agent-based social simulation: A computer science view*, Journal of Artificial Societies and Social Simulation, **5**(1).

[13] Kang, M., L.B. Waisel, and W.A. Wallace, 1998, *Team-Soar: A model of team decision making*, in Simulating Organizations: computational models of institutions and groups, M.J. Prietula, K.M. Carley, and L. Gasser, Editors, AAAI Press/The MIT Press.

[14] Lesser, V.R., 1999, *Cooperative multiagent systems: a personal view of the state of the art*, IEEE Transactions on Knowledge and Data Engineering, **11**(1): p. 133 - 142.

[15] Howard, A., M.J. Mataric, and G.S. Sukhatme, 2002, *Mobile sensor network deployment using potential fields: a distributed scalable solution to the area coverage problem*, in DARS 02, Fukuoka, Japan.

[16] Singh, M.P., A.S. Rao, and M.P. Georgeff, 2000, *Formal methods in DAI: logic-based representation and reasoning*, in Multiagent Systems: a modern approach to distributed artificial intelligence, G. Weiss, Editor, The MIT Press.

[17] Carley, K.M. and L. Gasser, 2000, *Computational organization theory*, in Multiagent Systems: a modern approach to distributed artificial intelligence, G. Weiss, Editor, The MIT Press.

[18] Castelfranchi, C., 2000, *Engineering social order*, in ESAW00, Berlin.

[19] Filipe, J., 2002, *A normative and intentional agent model for organization modeling*, in Third International Workshop Engineering Societies in the Agents World.

[20] Hollnagel, E., 1993, *Human reliability analysis: context and control*, Computers and People, London, Academic Press.

[21] Kirlik, A., 1998, *The design of everyday life environments*, in A Companion to Cognitive Science, W. Bechtel and G. Graham, Editors, Oxford, Blackwell, p. 702-712.

[22] Vicente, K.J., 1990, *A few implications of an ecological approach to human factors*, Human Factors Society Bulletin, **33**(11): p. 1 - 4.

[23] Vicente, K.J., 1999, *Cognitive Work Analysis: Towards Safe, Productive and Healthy Computer Based Work*, Lawrence Erlbaum.

[24] Decker, K.S., 1994, *Environment centered analysis and design of coordination mechanisms*, in Department of Computer Science, Amherst, University of Massachusetts.

[25] Shah, A.P. and A.R. Pritchett, 2004, *Work Environment Analysis: Environment centric multi-agent simulation for design of socio-technical systems*, in Joint Workshop on Multi-Agent and Multi-Agent Based Simulation: Third International Joint Conference on Autonomous Agents and Multi-Agent Systems, New York.

[26] Odell, J., et al., 2002, *Modeling agents and their environment*, in Agent Oriented Software Engineering (AOSE) III, F. Giunchiglia, J. Odell, and G. Weiss, Editors, Berlin, Lecture Notes on Computer Science, Springer, p. 16 - 31.

[27] Corker, K.M., 1994, *Man-machine integration design and analysis system (MIDAS) applied to a computer-based procedure-aiding system*, in 38th Annual Meeting of the Human Factors and Ergonomics Society, Santa Monica, CA: Human Factors and Ergonomics Society.

[28] Laughery, K.R. and K.M. Corker, 1997, *Computer modeling and simulation of human/system performance*, in Handbook of Human Factors, G. Salvendy, Editor, New York, John Wiley.

[29] Pritchett, A.R., S.M. Lee, and D. Goldsman, 2001, *Hybrid-system simulation for national airspace systems safety analysis*, AIAA Journal of Aircraft, **38**(5): p. 835-840.

[30] Niedringhaus, W.P., 2004, *The Jet:Wise model of national air space system evolution*, Simulation: Transactions of the Society for Modeling and Simulation, **80**(1): p. 45-58.

[31] Nair, R., et al., 2003, *Taming Decentralized POMDPs: Towards efficient policy computation for multi-agent settings*, in International Joint Conference on Artificial Intelligence.

[32] Pritchett, A.R., et al., 2002, *Examining air transportation safety issues through agent-based simulation incorporating human performance models*, in IEEE/AIAA 21st Digital Avionics Systems Conference, Irvine, CA.

[33] Borenstein, J., H.R. Everett, and L. Feng, 1996, *Navigating mobile robots: Systems and techniques*, AK Peters, Ltd.

[34] Schraagen, J.M., S.F. Chipman, and V.L. Shalin, eds., 2000, *Cognitive Task Analysis*, Mahwah, NJ, Lawrence Erlbaum Associates.

[35] Pritchett, A.R. and A.P. Shah, 2004, *Designing the work environment of socio-technical systems through agent-based modeling and simulation*, in IFAC/IFIP/IFORS/IEA Symposium on the Analysis, Design and Evaluation of Human Machine Systems, Atlanta.

[36] Law, A.M. and W.D. Kelton, 1999, *Simulation modeling and analysis*, 3rd ed, New York, NY, McGraw-Hill.

[37] Ghosh, S. and T.S. Lee, 2000, *Modeling and asynchronous distributed simulation: analyzing complex systems*, New York, IEEE Press.

[38] Lee, S.M., 2002, *Agent-based simulation of socio-technical systems: software architecture and timing mechanisms*, in Industrial and Systems Engineering, Atlanta, GA, Georgia Institute of Technology.